استخدام خوارزمية Bezier كمعيار لمقارنة لغتي C# و Python

زمزم محمد العاشوري[1]، عمر عبد الغني شيبة[1]، عامر صالح سهل[1]، أنور عباس عبد الله[2]

[1]قسم علوم الحاسوب- كلية تقنية المعلومات -جامعة سبها- ليبيا
[2]قسم الهندسة الالكترونية - كلية العلوم الهندسية والتقنية -جامعة سبها- ليبيا
البريد الالكتروني : zamzam.m92@gmail.com

الملخص

لكل لغة برمجة خصائص تميزها عن اللغات الأخرى وتجعلها مناسبة بدرجات متفاوتة لكل غرض من اغراض هذه اللغات، كما أن لغات البرمجة أيضا لها خصائص وحدود مشتركة بحكم أن كل هذه اللغات صممت للتعامل مع الحاسوب، حيث تتمتع كل لغة بتصميم خاص يختلف عن طريقة تصميم اللغات الأخرى في التعامل مع المعطيات، كما تختلف في طبيعة الطرق والتسهيلات التي توفرها اللغة للتعامل مع مشكلة معينة ، ومن هنا فإن هذه الدراسة تهدف الى اجراء مقارنة بين لغتين من لغات البرمجة والمتمثلة في C#، PYTHON من خلال استخدامهما في مجال الهندسة الحسابية بتطبيق خوارزمية Bezier للوقوف على اوجه الاختلاف والتباين بين أداء اللغتين وذلك من خلال عدد من معايير التقييم والمتمثلة في السرعة، واستخدام الذاكرة، و عدد الأسطر(loc).

الكلمات المفتاحية : الهندسة الحسابية، خوارزمية بيزير، المنحنيات، الأسطح، لغات البرمجة

Abstract

Each programming language has characteristics that distinguish it from other languages and make it suitable to varying degrees for each of the purposes of these languages, and programming languages also have common characteristics and limits by virtue of the fact that all these languages are designed to interact with the computer, as each language has a special design that differs from the way other languages are designed In dealing with data, they also differ in the nature of the methods and facilities provided by the language to deal with a specific problem, hence this study aims to make a comparison between two programming languages represented in C# and PYTHON through their use in the field of computational engineering by applying the Bezier algorithm to find out the differences and contrasts between the performance of the two languages through a number of evaluation criteria, which are speed, memory usage, and the number of lines (loc).

Keywords: computational geometry, Bezier algorithms, Curves ،surfaces ،Programming languages.

المقدمة

تعد المقارنة بين لغات البرمجة موضوعاً شائعاً بين المبرمجين ومطوري البرامج، مع التغييرات الأخيرة في معايير البرمجة والتحديثات المستمرة في تصميم الأجهزة، تم تطوير العديد من لغات البرمجة الجديدة، بينما تخضع اللغات الموجودة حالياً للعديد من التحسينات من حيث التصميم والتنفيذ، فكل لغة برمجة خصائصها التي تميزها عن الأخرى وتجعلها مناسبة بدرجات متفاوتة لكل نوع من أنواع البرامج وحسب المهمة المطلوبة من هذه البرامج، كما أن للغات البرمجة أيضا خصائص وحدود مشتركة [1].

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية

ترتكز رسومات الحاسب على نماذج ضمنية للمشاهد التي نراها، والتي هي عبارة عن تمثيلات رياضية ثلاثية الأبعاد لشكل وخصائص المشهد المطلوب والتي تخضع للمعالجة من قبل الحاسب. حيث ان هذه المشاهد تتم نمذجتها في الحاسب إلى أشكال رياضية، ويعبر عنها في ذاكرة الحاسب بحسب درجة تعقيد المشهد، فأبسط الأشكال هي الوحدات الأولية والتي تتضمن الأشكال الهندسية التقليدية التي نعرفها، كالمكعب، والأسطوانة، والكرة والمخروط، ثم تأتي المضلعات في المرتبة الثانية، والتي تعتبر أحد أبرز الأساليب الكلاسيكية المتبعة في نمذجة الأشكال في الفضاءات ثلاثية الأبعاد، وأخيرا المنحنيات والسطوح الملساء من أجل تمثيل السطوح والأشكال المعقدة[2].

غالباً ما تظل المناقشات حول فعالية لغات البرمجة المختلفة غير حاسمة بسبب ندرة البيانات وعدم وجود مقارنات مباشرة، يعد الأداء وقابلية التوسع من العوامل الكبيرة في تطوير البرامج، وغالباً ما يكون اختيار لغة البرمجة الصحيحة عملية طويلة، فغالباً ما تختار الشركات لغات ذات مستوى أعلى مثل python أو JavaScript نظراً لسهولة استخدامها ومرونتها الكبيرة، ولكن عندما يتعلق الأمر بالأداء، يُفضل دائماً استخدام لغة منخفضة المستوى. حيث تحتاج بعض التطبيقات إلى التعامل مع الكثير من إدخال وإخراج الملفات بشكل فعال، وقد يحتاج البعض الآخر إلى معالجة البيانات بطرق مختلفة، وقد يقوم البعض بالأمرين معاً.

من هنا ستجيب هذه الدراسة من خلال الاقسام التالية عن التساؤل عن اوجه الاختلاف بين أداء لغة C# و لغة Python عند رسم المنحنيات والاسطح باستخدام منحنى Bezier ، أما بقية هذه الورقة فهي منسقة كالتالي: القسم الثاني يستعرض بشكل مختصر بعض الدراسات ذات العلاقة، أما المنهجية المتبعة في هذه الدراسة فتم وصفها في القسم الثالث، أما بعض المصطلحات الأساسية ذات العلاقة فقد تم استعراضها في القسم الرابع، وأما القسم الخامس فقد خصص للتجارب ومناقشة النتائج، بينما كان القسم السادس للخلاصة.

دراسات ذات علاقة

تم القيام بالعديد من الأعمال في مجال مقارنة لغات البرمجة، حيث قدم Akram [3] دراسة لمقارنة لغات البرمجة المختلفة من خلال دراسة معايير محددة وهي: السرعة، واستخدام الذاكرة، وحجم الملف التنفيذي، وحجم الشفرة المصدرية، حيث ركزت هذه الدراسة على أربع لغات لمعرفة أفضل لغة لكل معيار من المعايير المختارة عند استخدامها في رسومات الحاسوب والهندسة الحسابية.

كما قدم Oguntunde, Bosede Oyenike [1] مقارنة تحليلية لستة لغات برمجة اثنتين من كل نوع من لغات البرمجة العلمية وغير العلمية ولغات البرمجة الشيئية باستخدام خوارزمية لأداء الجمع والتبديل لتنفيذ المقارنة، حيث تم اختبار معيارين فقط وهما استهلاك الذاكرة ومتطلبات زمن التشغيل.

قدم أيضا Goyal [5] دراسة مقارنة لأربع لغات برمجة،C و Java و #C و Python ، حيث تمت المقارنة باستخدام ثلاثة معايير متمثلة في استهلاك الذاكرة، واستخدام وحدة المعالجة المركزية، و زمن التنفيذ .

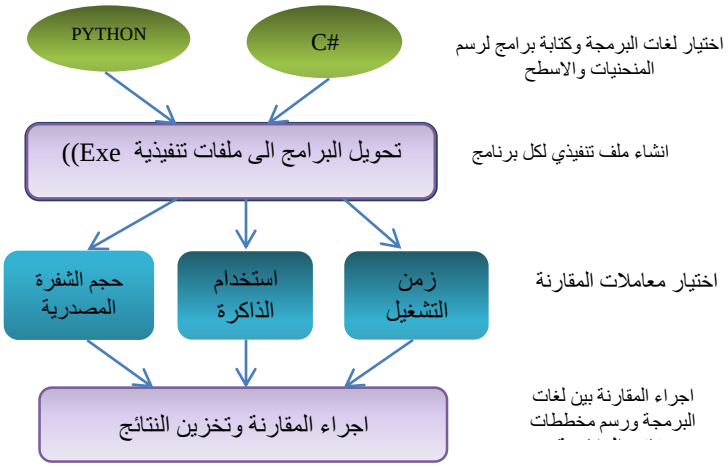
ومن هنا فإن هذه الدراسة تسلط الضوء على نوع اخر من المعايير لمقارنة لغات البرمجة والمتمثل في تطبيق خوارزمية بيزير في مجال الهندسة الحسابية لرسم المنحنيات والاسطح .

المنهجية

للإجابة على سؤال الدراسة أجريت العديد من التجارب استخدمت فيها القياسات الكمية لمقارنة لغتي البرمجة المستهدفة من المقارنة من حيث سرعة التنفيذ واستخدام الذاكرة. وعدد الاسطر، حيث يتم أولا كتابة البرنامج لرسم المنحنيات والاسطح بلغة البرمجة المستهدفة ومن ثم تحويل البرنامج الى صيغة تنفيذية وبعدها يتم اختيار المعايير

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية

لاجراء عملية المقارنة، ولانجاز هذه المراحل فقد تم بناء تطبيق لتحقيق هذا الغرض، الشكل (1) يوضح مراحل المنهجية المتبعة في هذه الدراسة.



شكل (1) مراحل المنهجية المتبعة في الدراسة

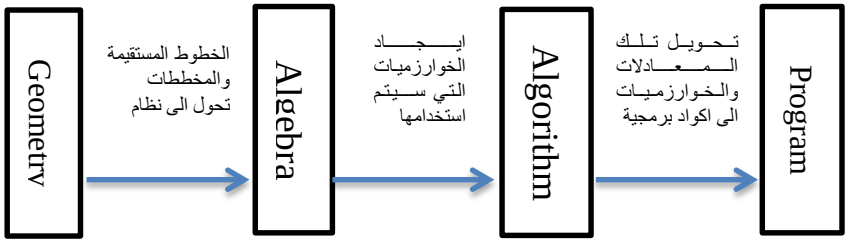
مصطلحات ذات علاقة

في هذا القسم سنستعرض بعض المصطلحات ذات العلاقة بموضوع الدراسة والتي اهمها:

- الهندسة الحسابية

الهندسة الحسابية هي مجال رياضي يتضمن تصميم وتحليل وتنفيذ خوارزميات فعالة لحل مشاكل المدخلات والمخرجات الهندسية، حيث يتم استخدامها للتعرف على الأنماط ووصف خوارزميات النمذجة المستخدمة في معالجة المنحنيات والأسطح[6][8][9].

تم تطوير الهندسة القائمة على الحساب من خلال دراسة خوارزميات الفرز والبحث المستخدمة في المساحات أحادية البعد لحل المشكلات التي تنطوي على المدخلات متعددة الأبعاد، كما ساهم تطورها أيضًا في الحصول على رؤى من نظريات الرسم البياني الحسابي المستخدمة في الهندسة العادية او الطبيعية [3]. الشكل (2) يوضح التسلسل العام لمفهوم الهندسة الحاسبية.



شكل (2) التسلسل العام لمفهوم الهندسة الحسابية

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية

حيث ان المشاكل الأساسية التي تدرسها الهندسة الحسابية هي تمثيل ونمذجة الاسطح، أما أهم الأدوات فتتمثل في المنحنيات والأسطح البارامترية، مثل منحنيات بيزيير، ومنحنيات الشريحة والأسطح.

- منحنى **Bezier**

المنحنى هو كائن رياضي يتكون من مجموعة من النقاط المتجاورة كخط، ويكون الخط المستقيم حالة خاصة من المنحنى، وقد تكون المنحنيات ثنائية الابعاد (المنحنيات في المستوى) أو ثلاثية الابعاد (المنحنيات في الفراغ (، حيث يعتبر منحنى بيزير(Bézier) منحنى بارامتري يستخدم دوال متعددات الحدود كأساس [10].

حيث تمثل منحنيات Bézier شكل من اشكال الدوال البارامترية، تعمل هذه المنحنيات بنقطة بداية ونقطة نهاية مع انحناءات تتأثر بواحدة أو أكثر من نقاط التحكم، حيث يتم تشكيل المنحنى باستخدام نقاط التحكم، بحيث يؤدي تحريك نقطة من نقاط التحكم الى تغيير شكل المنحنى، يُطلق على المضلع الذي تم الحصول عليه عن طريق توصيل جميع نقاط التحكم باسم مضلع التحكمcontrol polygon [11].

يتم تعريف منحنى بيزير العام للدرجة n + 1 بواسطة نقاط التحكم $P_0,P_1,\ldots,P_n$ كما في المعادلة (1) والتي تساوي

$$B(t) = \sum_{i=0}^{n} Pibi\ ,n(t), 0 \le t \le 1 \quad (1)$$

حيث bi هي دالة بيرنشتاين متعددة الحدود للدرجة n المعرفة بواسطة$b_{i,n}(t)$ كما في المعادلة (2):

$$b_{i,n}(t) = \binom{n}{i} (1-t)^{n-1} \quad (t)_i \qquad (2)$$

حيث $\binom{n}{i}$ المعامل ذو الحدين المعرف في المعادلة (3)

$$(n\ i\ ) = \frac{n!}{(n-i)!i!} \quad (3)$$

- اسطح **Bezier**

تعد أسطح Bezier واحدة من أكثر العناصر الأولية – المستخدمة في النمذجة عالية الجودة في أدوات CAD / CAM وبرامج الرسومات. فقد ساهمت الخصائص الرياضية للخوارزمية مع التطبيقات الصناعية الناجحة، في زيادة شعبية تمثيل بيزير[12]، حيث يتم تعريف سطح Bezier كما في المعادلة (4) :

$$S(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} P_{i,j} B_{i,m}\ (u) B_{j,n}(v) \quad (4)$$

حيث

$$B_{i,m}(u) = \binom{m}{i} (1-u)^{m-i}\ u^i \qquad 0 \le i \le\ m$$

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية

- **لغات البرمجة**

لغة البرمجة هي مجموعة من التعليمات الشبيهة باللغة الإنجليزية والتي تتضمن مجموعة من القواعد (بناء الجملة) لوضع التعليمات معاً لإنشاء أوامر لغة البرمجة التي هي اللغة الرسمية للحاسوب مصممة لإيصال التعليمات إلى الجهاز. حيث يمكن استخدام لغات البرمجة لإنشاء برامج للتحكم في سلوك الآلة أو للتعبير عن الخوارزميات. التطور في تكنولوجيا لغات البرمجة سريع أيضًا. منذ ظهور الجيل الأول من لغات البرمجة في أوائل عام 1950 ، فإن لغات البرمجة لها خمسة أجيال. يهدف الجيل الأخير من لغات البرمجة إلى "جعل الحاسوب يحل مشكلة معينة دون الحاجة للمبرمجين. ومع ذلك، وفقًا لمؤشر مجتمع البرمجة TIOBE الذي يعطي مؤشرًا على شعبية لغات البرمجة، فإن الجيل الثالث من لغات البرمجة لا يزال يستخدم على نطاق واسع[13]، في هذه الدراسة تم اختيار لغتين من هذه اللغات شائعة الاستخدام والمتمثلة في:

لغة البرمجة# **C:** وهي لغة برمجة متعددة النماذج وموجهة للكائنات والتي تسهل الوراثة والتجريد وتعدد الأشكال والتغليف، تزداد شعبيتها بسبب كفاءتها وسهولة ترميزها. يدير الإطار تنفيذ التطبيقات وخدمات الويب، بالإضافة إلى ذلك، فإنه يوفر العديد من الوظائف الأخرى التي تشمل إدارة الذاكرة وفرض الأمن. مثل لغة Java، تحتوي لغة #C أيضًا على مجموعة تلقائية للقمامة ولها بنية بناء مماثلة.

لغة البرمجة **PYTHON** : ظهرت هذه الغة لأول مرة عام 1991 بعد اطلاقها للبرمجة من قبل مخترعها الهولندي (Rossum Van Guido)، حيث تمتاز هذه اللغة بأنها تركز على قابلية القراءة(Readability) و ذلك لاستخدامها كلمات مفتاحية مشابهة للغة البشر تسهل قراءتها من قبل المختصين وغيرهم، كما أن من مميزاتها المرونة العالية في استخدام المتغيرات بدون اعلان وهذا من شأنه أن يقلل وقت البرمجة والتطوير الى الحد الأدنى، كغيرها من لغات المستوى العالي، كما تعتبر لغة بايثون بأنها لغة موجهة للكائنات (object oriented) وتسمح بتطوير تطبيقات متعددة الأغراض وتطوير مكتبات النظام القياسية.

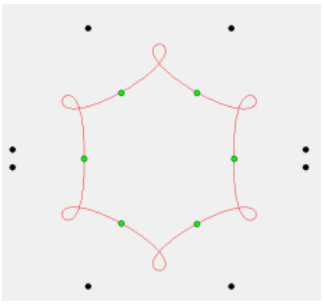
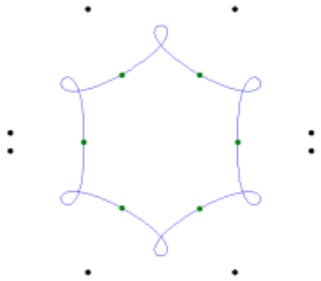
التجارب ومناقشة النتائج

تم في هذه الدراسة اختيار أثنين من لغات البرمجة والمتمثلة في لغة ,C# ولغة PYTHON كما تم اختبار ثلاثة معاملات، لفحص هذه الغات للوصول الى السمات والقدرات المحددة لكل لغة، حيث تمثلت هذه المعاملات في الاتي:

- سرعة التنفيذ : والتي تعني سرعة تطبيق البرنامج وتنفيذ التعليمات داخله وتعتبر من أهم العوامل التي تؤخذ بعين الاعتبار عند تقييم لغات البرمجة. [14]
- استخدام الذاكرة: حيث تتطلب معالجة الصور قدراً كبيراً من الذاكرة لتخزين هياكل البيانات الكبيرة متعددة الأبعاد من نوع المصفوفة الموجودة في مثل هذه التطبيقات، لذلك يعتبر هذا المعيار مهم في عملية مقارنة استخدام الذاكرة بالنسبة للغات البرمجة.
- Line of Code (LOC) : وهو مقياس برمجيّ أو أداة قياس برمجية تُستخدم لقياس حجم البرنامج عبر احتساب عدد الأسطر الموجودة في نصّ البرنامج. حيث تُستخدم هذه الشيفرة عادةً لتوقّع حجم الجهد المطلوب لتطوير برنامج معيّن وتقدير الحجم الإنتاجي للبرنامج وإمكانية صيانته وتعديله بعد إنتاجه.

تم اجراء التجارب على لغتي البرمجة وذلك لرسم منحنيات واسطح بيزير، فكانت التجربة الاولى لرسم منحنى بيزير من الدرجة الرابعة والذي تظهر نتائجه في الشكل (6) حيث يحتوي على 600 نقطة .

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية

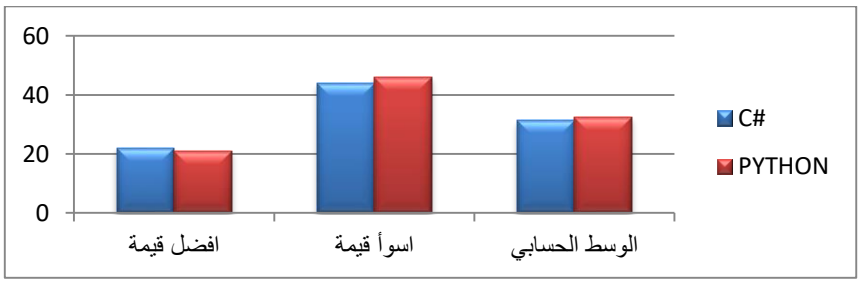
PYTHON	C#
	

شكل (6) رسم منحنى بيزير بلغة C# و Python

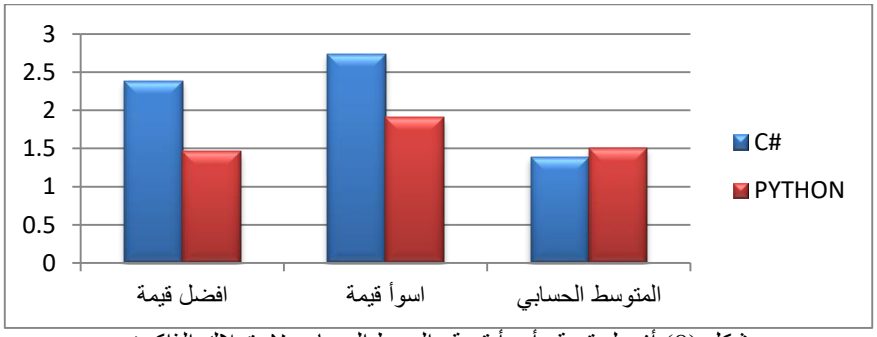
الجدول (1) يصف النتائج المتحصل عليها من حيث أفضل قيمة وأسوأ قيمة والوسط الحسابي لسرعة التنفيذ واستهلاك الذاكرة للمنحنيات، والمبينة في الشكل البياني (7) والشكل البياني(8).

جدول(1) افضل قيمة وأسوأ قيمة والوسط الحسابي للنتائج المتحصل عليها لمعاملي سرعة التنفيذ واستهلاك الذاكرة

Time & Memory	Time in Python (ms)	Time in C#(ms)	Memory usage in Python (kb)	Memory usage in C# (kb)
Best Case	21	22	1.457031	2.386719
Worst Case	46	44	1.910156	2.738281
Average Case	32.7	31.3	1.505469	2.588672



شكل (7) أفضل قيمة وأسوأ قيمة والوسط الحسابي لسرعة التنفيذ



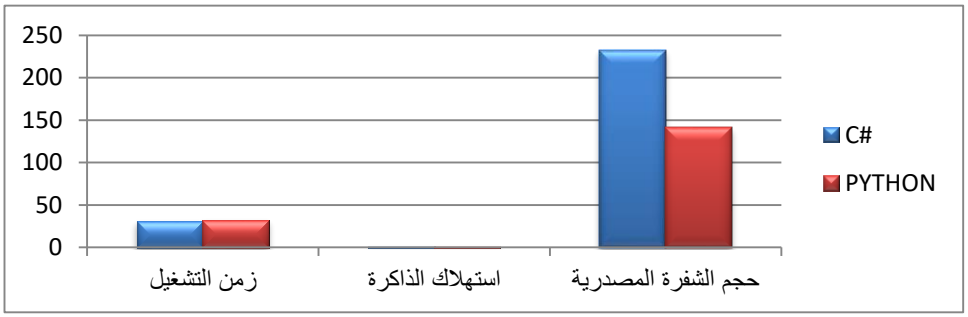
شكل (8) أفضل قيمة وأسوأ قيمة والوسط الحسابي لاستهلاك الذاكرة

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية

كما يصف الجدول (2) نتائج المتوسط الحسابي لزمن التشغيل واستهلاك الذاكرة، وحجم الشفرة المصدر للمنحنيات، والموضحة بالشكل البياني (9).

جدول (2) نتائج المتوسط الحسابي لزمن التشغيل واستهلاك الذاكرة، وحجم الشفرة المصدرية للمنحنيات

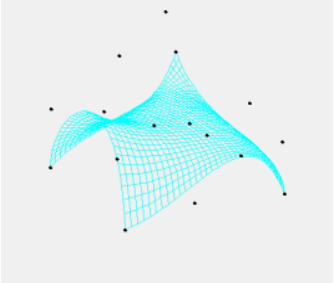
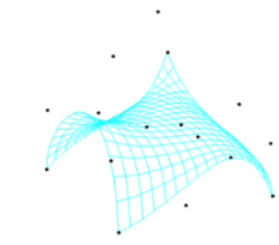
Languages Criteria	PYTHON	C#
Time in seconds	32.5	31.3
Memory usage in Kb	1.505469	2.588672
Source code size (LOC)	141	233



شكل (9) الرسم البياني لنتائج المتوسط الحسابي لزمن التشغيل واستهلاك الذاكرة، وحجم الشفرة المصدر للمنحنيات.

من خلال النتائج المسجلة في الجدول(2)، يمكن ملاحظة الفرق في زمن التشغيل، حيث ان لغةC# هي الاسرع و استغرقت زمن (31.3 ملي ثانية) لتنفيذ الرسم، بينما استغرقت لغة Python زمن(32.5 ملي ثانية)، اما فيما يتعلق باستهلاك الذاكرة ، يتضح أن هناك فرق ضئيل بين اللغتين كان اداء لغة Python أفضل حيث استهلكت لغة Python (1.5 كيلو بايت) بينما استغرقت لغة C# (2.5 كيلو بايت)، نلاحظ ايضا ان عدد اسطر الشفرة المصدرية للغة C# اكثر من عدد أسطر الشفرة المصدرية للغة Python.

اما التجربة الثانية فكانت لرسم سطح Bezier من الدرجة الرابعة والمبين في الشكل (9) حيث يحتوي على 25600 نقطة .

PYTHON	C#
	

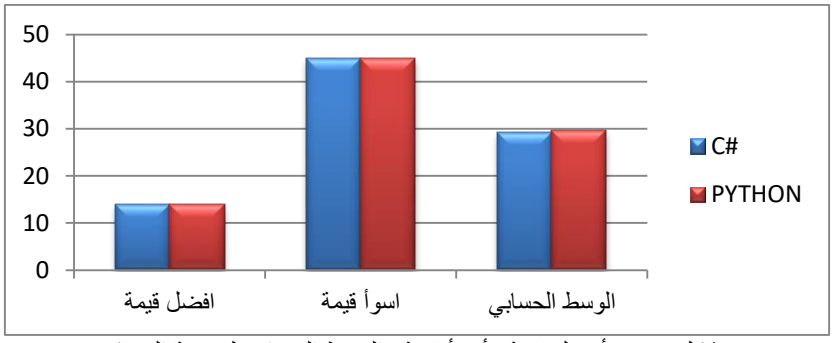
شكل (9) رسم الاسطح بلغة C# و Python

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية

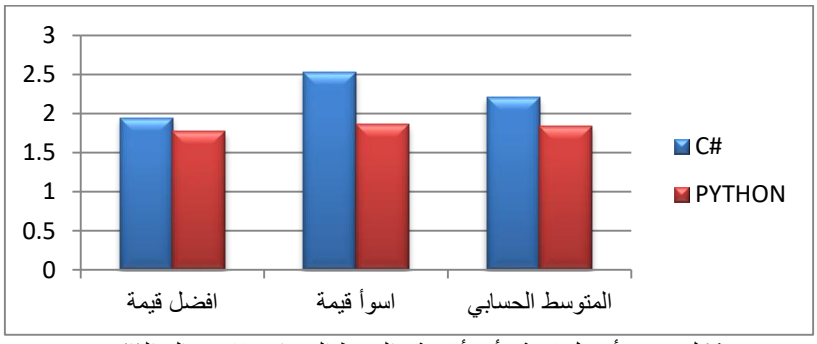
الجدول (3) يصف النتائج المتحصل عليها من حيث أفضل قيمة وأسوأ قيمة والوسط الحسابي لسرعة التنفيذ واستهلاك الذاكرة للأسطح، والمبينة في الشكل البياني (10)،والشكل البياني(11).

جدول (3) افضل قيمة وأسوأ قيمة والوسط الحسابي للنتائج المتحصل عليها من قياس سرعة التنفيذ واستهلاك الذاكرة للأسطح

Time & Memory	Time in Python (ms)	Time in C#(ms)	Memory usage in Python (kb)	Memory usage in C#(kb)
Best Case	14	14	1.773438	1.941406
Worst Case	45	45	1.859375	2.527344
Average Case	29.7	29.2	1.839844	2.209766



شكل (10) أفضل قيمة وأسوأ قيمة والوسط الحسابي لسرعة التنفيذ



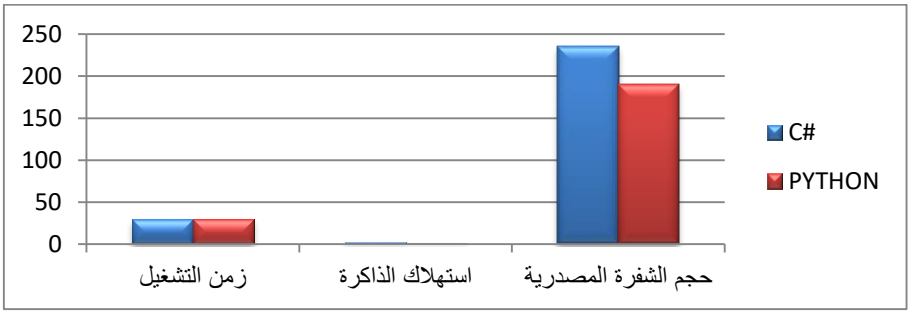
شكل (11) أفضل قيمة وأسوأ قيمة والوسط الحسابي لاستهلاك الذاكرة

الجدول (4) يصف نتائج المتوسط الحسابي لزمن التشغيل واستهلاك الذاكرة، وحجم الشفرة المصدرية للأسطح، بينما يوضح الشكل البياني (12) هذه النتائج.

جدول (4) نتائج المتوسط الحسابي لزمن التشغيل واستهلاك الذاكرة ، وحجم الشفرة المصدر لرسم الأسطح

Languages Criteria	PYTHON	C#
Time in seconds	29.7	29.2
Memory usage in Kb	1.839844	2.209766
Source code size *(LOC)*	190	235

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية



شكل (12) نتائج المتوسط الحسابي لزمن التشغيل واستهلاك الذاكرة، وحجم الشفرة المصدر لرسم الأسطح.

من خلال النتائج المدرجة في الجدول(4)، نلاحظ وجود تباين في النتائج بين التجربة الاولى والثانية، حيث انه عند رسم الاسطح كانت لغة Python هي الاسرع، حيث استغرقت زمن (26.3ملي ثانية) للرسم، بينما استغرقت لغة #C زمن(30.7 ملي ثانية)، يتضح أن هناك فرق ضئيل بين اللغتين في زمن التشغيل حيث استغرقت #C زمن (29.2ملي ثانية) للرسم، بينما استغرقت لغة Python زمن(29.7 ملي ثانية)، فيما يتعلق باستهلاك الذاكرة كان اداء لغة Python أفضل حيث استهلكت لغة Python (1.83 كيلو بايت) بينما استهلكت لغة #C (2.20 كيلو بايت)، اما بالنسبة لعدد الاسطر للشفرة المصدرية يتضح ان لغة #C لها عدد اسطر اكبر من لغة Python.

الخلاصة:

من خلال التجارب التي أجريت وتنفيذ خوارزمية بيزير لرسم المنحنيات والاسطح بأستخدام لغة **#C** و لغة **Python** وبناء على مجموعة المقاييس المختارة لعملية المقارنة فقد تبين ان لغة **Python** كانت الافضل في رسم المنحنيات والاسطح من حيث استهلاك الذاكرة وحجم الشفرة المصدرية، بينما تميزت لغة **#C** بسرعة التنفيذ في الوقت الذي تميزت فيه لغة **Python** ببساطة برامجها كتابتا وفهما، ومن هنا نستطيع القول بأن أداء لغات البرمجة يختلف من لغة الى اخرى بناء على مجال التطبيق وكذلك معايير القياس لهذا الاداء.

التوصيات:

بناء على محدودية هذه الدراسة من حيث عدد اللغات والمعايير المستهدفة في التجارب التي أجريت فإننا نوصي بإجراء تجارب على لغات اخرى وكذلك دراسة لبقية المعايير التي لم يتم دراستها بالاضافة الى دراسة خوارزميات اخرى في مجال الهندسة الحسابية.

المراجع

[1]. B. O. Oguntunde, “Comparative analysis of some programming languages,” *Transnatl. J. Sci. Technol.*, vol. 2, no. 5, pp. 107–118, 2012.
[2]. F. Durand, “A short introduction to computer graphics,” *MIT Lab. Comput. Sci.*, vol. 1, pp. 1–7, 2010.
[3]. M. Z. Akram, “Framework for Evaluating Programming Languages for Computer Graphics,” Universiti Putra Malaysia, 2003.
[4]. L. Prechelt, “An Empirical Comparison of Seven Programming Languages,” *Univ. Karlsruhe*, 2000.
[5]. P. Goyal, “Comparative Study of C, Java, C# and Jython,” 2014.

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية

[6]. M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, "Computational Geometry," 2008.
[7]. Zehra, F., Javed, M., Khan, D., & Pasha, M.. Comparative analysis of C++ and Python in terms of memory and time, (2020).
[8]. D. M. Mount, "CMSC 754 Computational geometry," *Lect. Notes, Univ. Maryl.*, pp. 1–122, 2002.
[9]. A. Selimi and M. Saracevic, "Computational geometry applications," *Southeast Eur. J. Soft Comput.*, vol. 7, no. 2, 2018.
[10]. M. Syakir and B. Saffie, "Bezier Curve Interpolation on Road Design." 2019, [Online]. Available: http://eprints.usm.my/48176/1/MOHD SYAKIR BIN SAFFIE cut.pdf.
[11]. Saffie, M. S., & Ramli, A. (2018). Bezier curve interpolation on road map by uniform, chordal and centripetal parameterization. AIP Conference Proceedings, 1974(1), 20087.
[12]. R. Concheiro, M. Amor, M. Bóo, and E. J. Padrón, "Free adaptive tessellation strategy of Bézier surfaces," in *2014 International Conference on Computer Graphics Theory and Applications (GRAPP)*, 2014, pp. 1–9.
[13]. J. O. Ogala and D. V Ojie, "Comparative analysis of c, c++, c# and java programming languages," *GSJ*, vol. 8, no. 5, 2020.
[14]. N. M. Patrikalakis and T. Maekawa, *Shape interrogation for computer aided design and manufacturing*, vol. 15. Springer, 2002.

Copyright © ISTJ
حقوق الطبع محفوظة
للمجلة الدولية للعلوم والتقنية